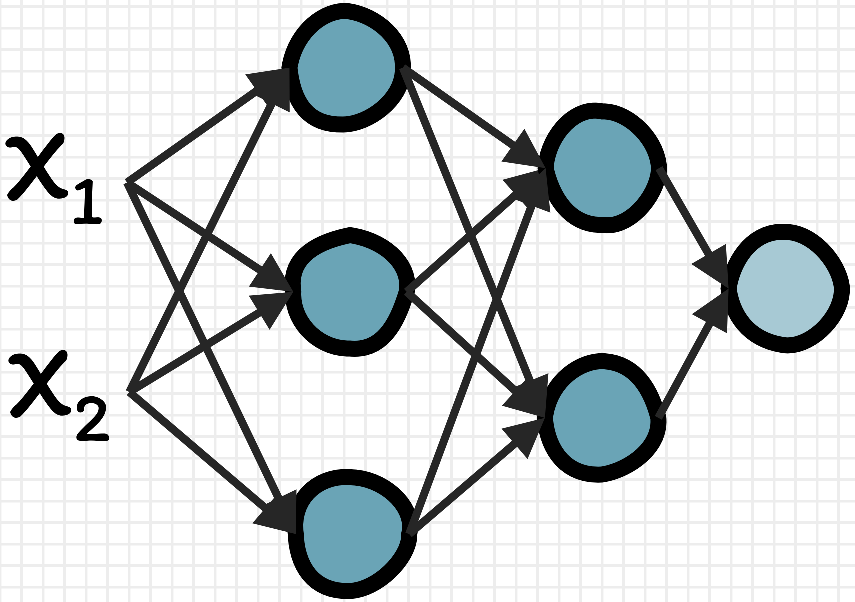
AIBYNETO

คอร์สสั้น ฟรี

Neural Networks

สำหรับผู้เริ่มต้น



Goals & Outlines

Goals:

- What is the most fundamental principle that underlies how NNs work?
- How are they trained to perform such tasks?

Outlines:

1. Well-known applications of Neural Networks (NNs)
2. Learning complex structures from data
3. Intro to NNs
4. Logistic regression
5. Gradient descent intuition
6. Backpropagation

1. Well-known applications of NNs

AIBYNETO

“Large Language Models (LLMs)”

ChatGPT | Gemini | LLaMA by Meta

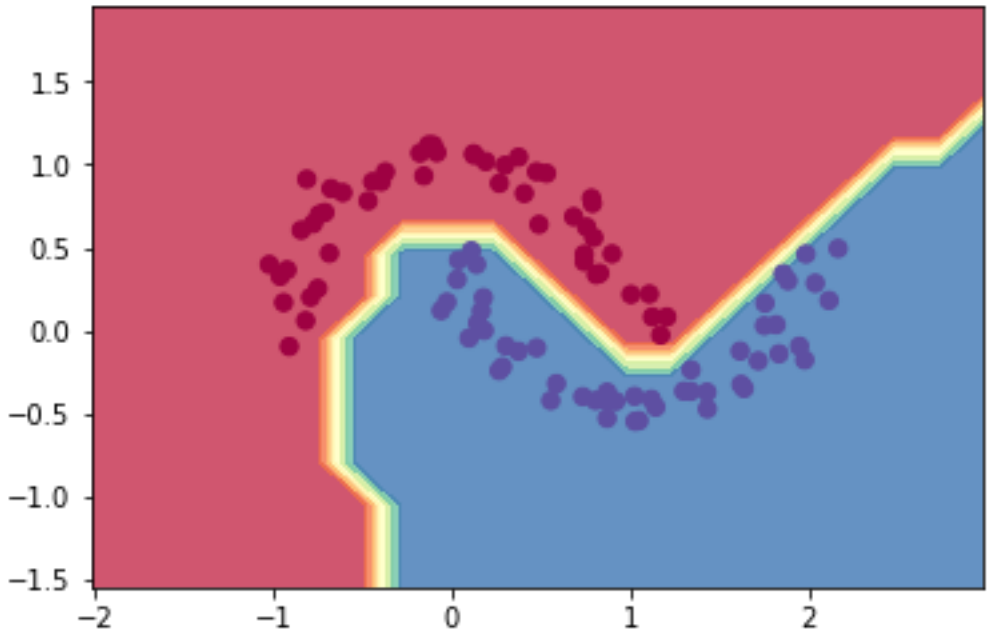
OpenAI | Google | Meta

1. Well-known applications of NNs

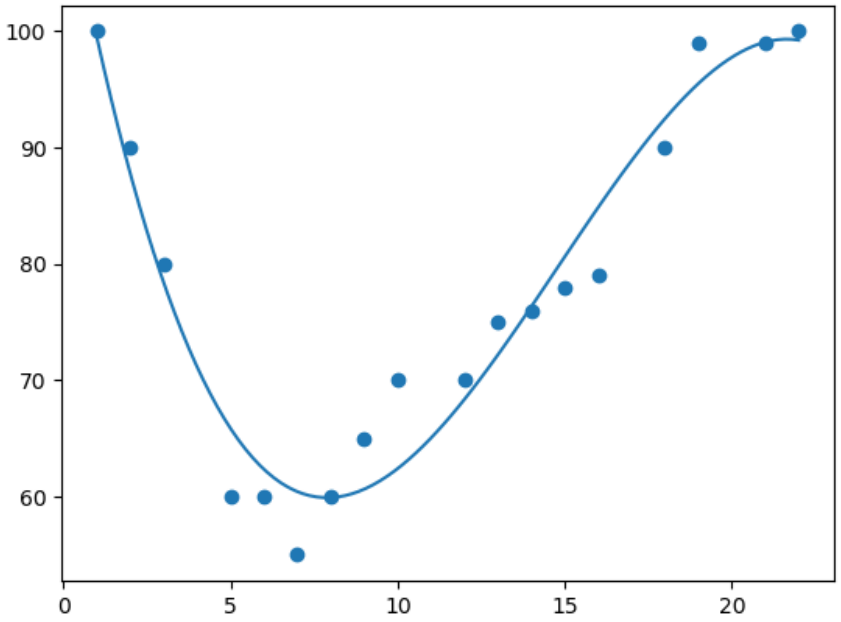
Example: Generative pre-trained transformer (GPT)

Transformer: Neural network architecture that solves sequence-to-sequence (time-series) tasks while handling long-range dependencies...

2. Learning complex structures from data



Classification



Regression

2. Learning complex structures from data

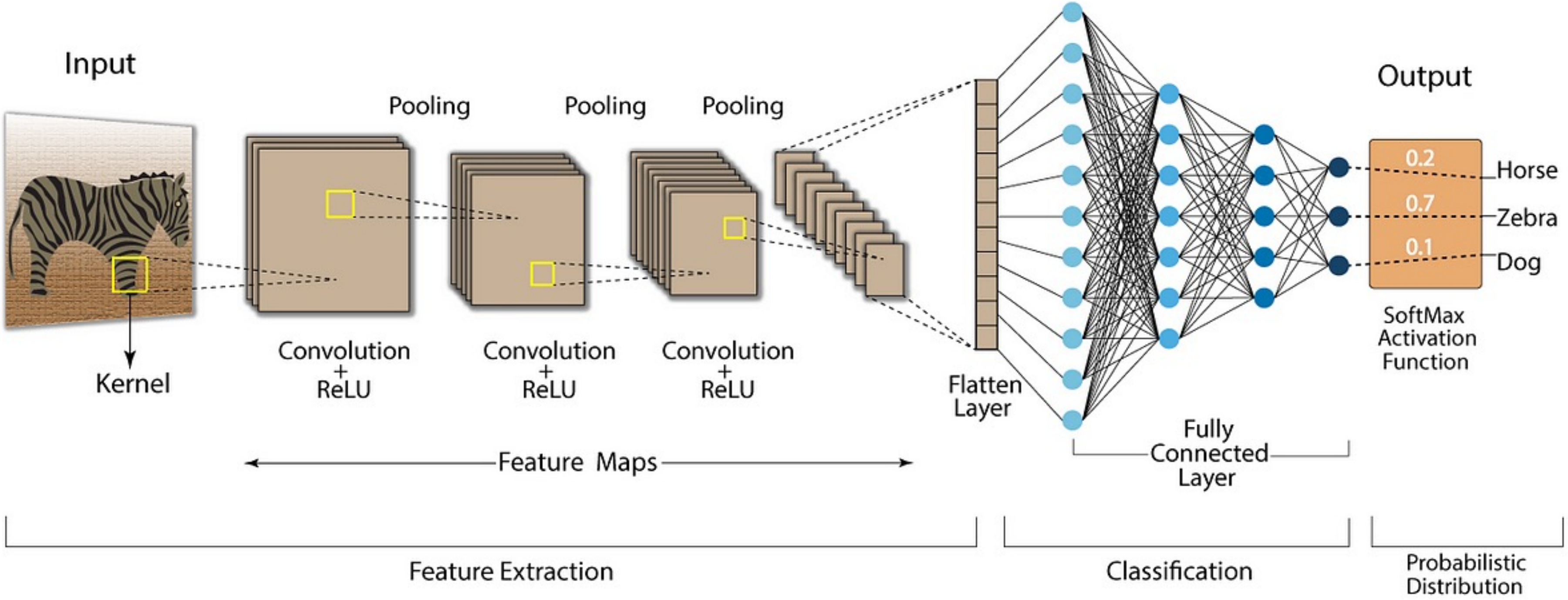
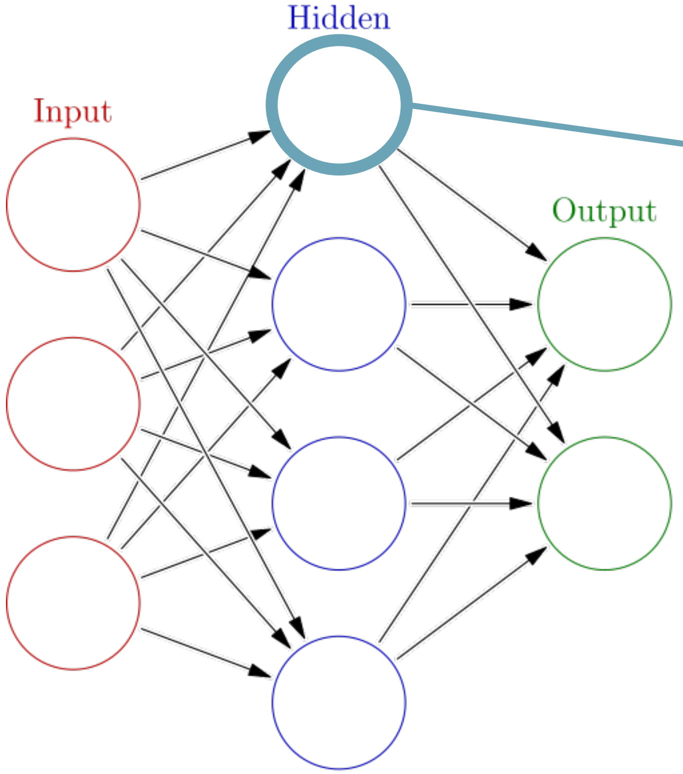


Image classification

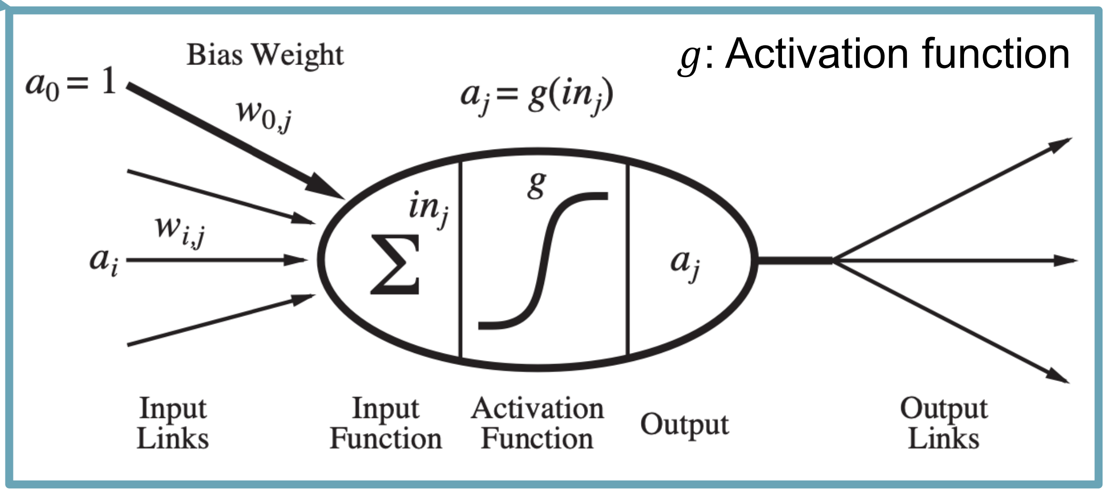
3. Intro to NNs

- A mathematical model of the brain
- Formulated by linear transformations + activation functions (to learn nonlinearities)



$$in_j = \sum_{i=0}^{n} w_{i,j}\, a_i$$

linear transformation



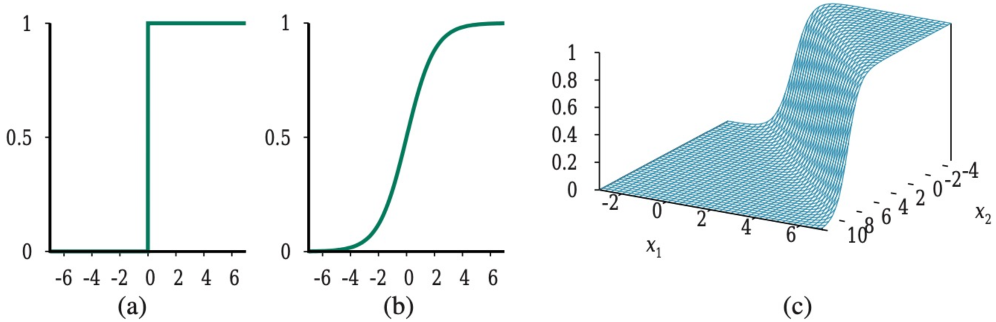
3. Intro to NNs

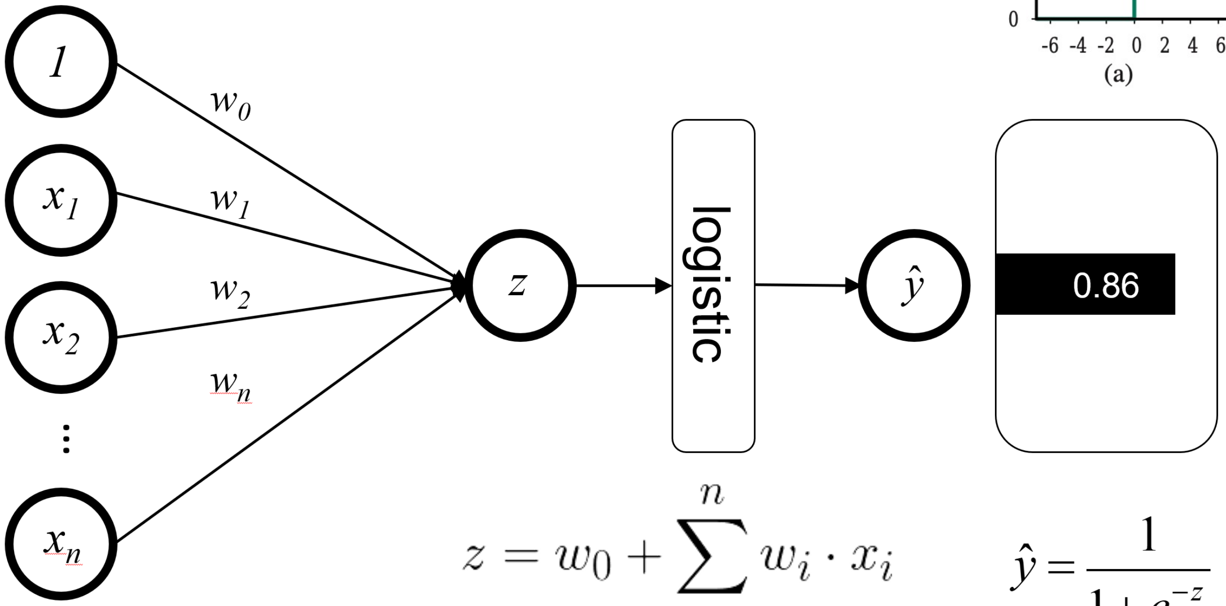
AIBYNETO

- Update $w_{i,j}$ such that we can successfully perform our task (could be a regression or classification task).
- What does it mean to succesfully perform something???
- Minimize loss(es)/cost

4. Logistic regression

1 neuron: Logistic regression!





$$z = w_0 + \sum_{i=1}^{n} w_i \cdot x_i$$

$$\hat{y} = \frac{1}{1+e^{-z}}$$

(b) and (c)

4. Logistic regression

AIBYNETO

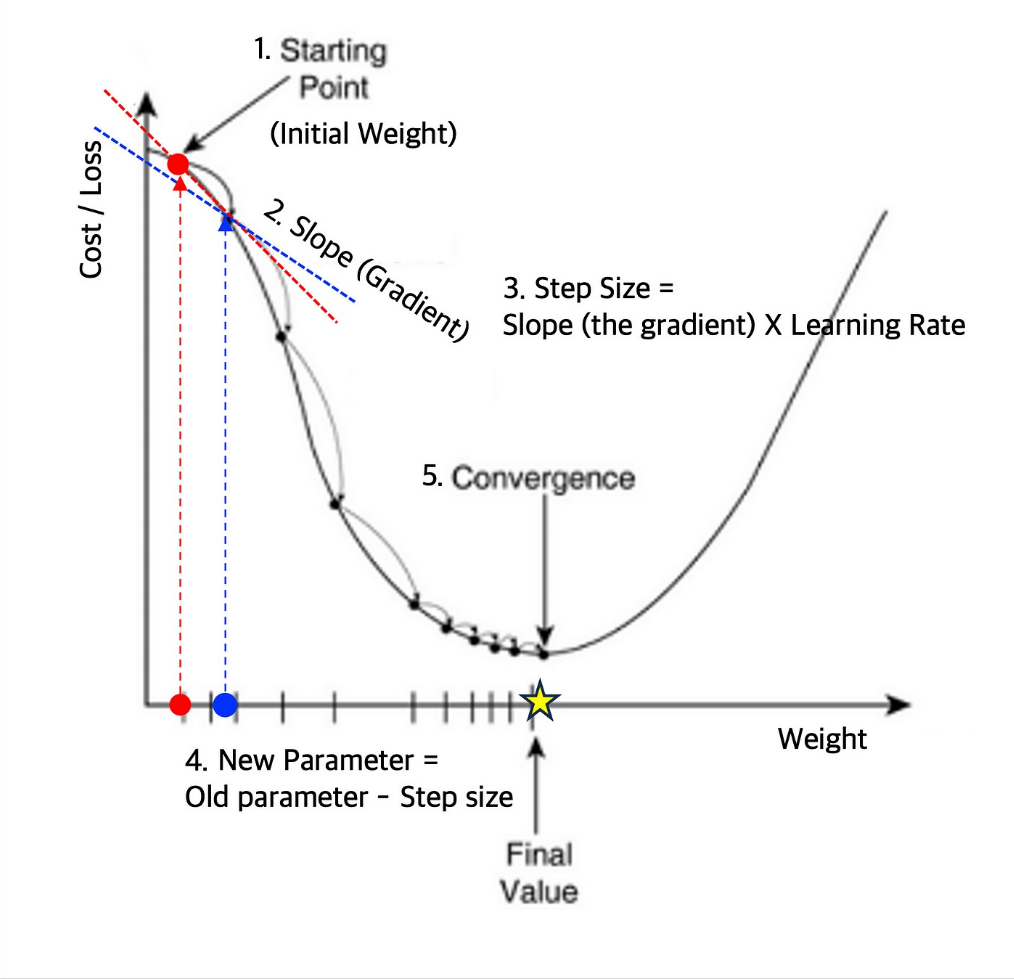
- 1 neuron: Logistic regression!
- With l2-loss (for simplicity), we know how to differentiate the loss function with respect to a weight. g is a sigmoid function.

$$h_{\mathbf{w}}(\mathbf{x}) = g(h_{\mathbf{w}}(\mathbf{x}))$$

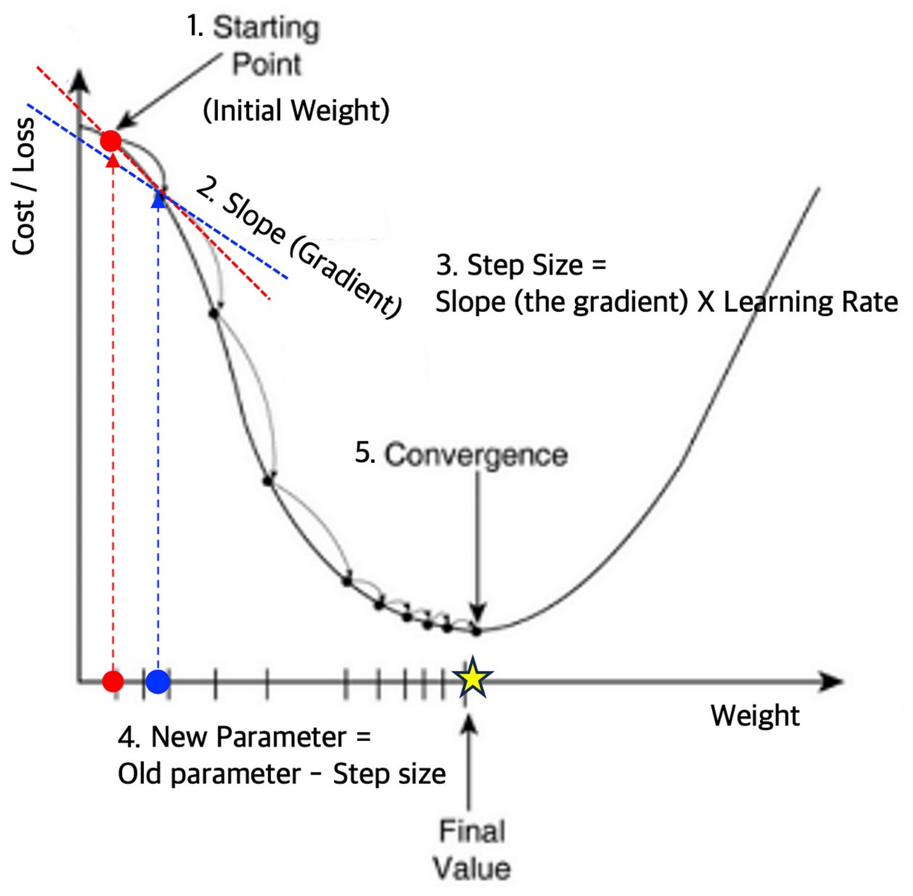
Hypothesis function

$$\begin{aligned}
\frac{\partial}{\partial w_i} Loss(\mathbf{w}) &= \frac{\partial}{\partial w_i}(y - h_{\mathbf{w}}(\mathbf{x}))^2 \\
&= 2\,(y - h_{\mathbf{w}}(\mathbf{x})) \times \frac{\partial}{\partial w_i}(y - h_{\mathbf{w}}(\mathbf{x})) \\
&= -2\,(y - h_{\mathbf{w}}(\mathbf{x})) \times g'(\mathbf{w}\cdot\mathbf{x}) \times \frac{\partial}{\partial w_i}\mathbf{w}\cdot\mathbf{x} \\
&= -2\,(y - h_{\mathbf{w}}(\mathbf{x})) \times g'(\mathbf{w}\cdot\mathbf{x}) \times x_i
\end{aligned}$$

5. Gradient descent intuition



5. Gradient descent intuition



$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Update rule:

$$\boxed{w \leftarrow w - \alpha \frac{dL}{dw}}$$

(how to walk in a loss landscape)

5. Gradient descent intuition

- 1 neuron: Logistic regression!
- **Training logistic regression with gradient descent!**

Derivative of the sigmoid function (Proof?):

$$g'(\mathbf{w} \cdot \mathbf{x}) = g(\mathbf{w} \cdot \mathbf{x})(1 - g(\mathbf{w} \cdot \mathbf{x})) = h_{\mathbf{w}}(\mathbf{x})(1 - h_{\mathbf{w}}(\mathbf{x}))$$

Update rule:

$$w_i \leftarrow w_i + \alpha(y - h_{\mathbf{w}}(\mathbf{x})) \times h_{\mathbf{w}}(\mathbf{x})(1 - h_{\mathbf{w}}(\mathbf{x})) \times x_i$$

6. Backpropagation

- Imagine that you know **how to efficiently differentiate the loss function with respect every parameters/weights** of your network.

- The backpropagation algorithm is all you need!

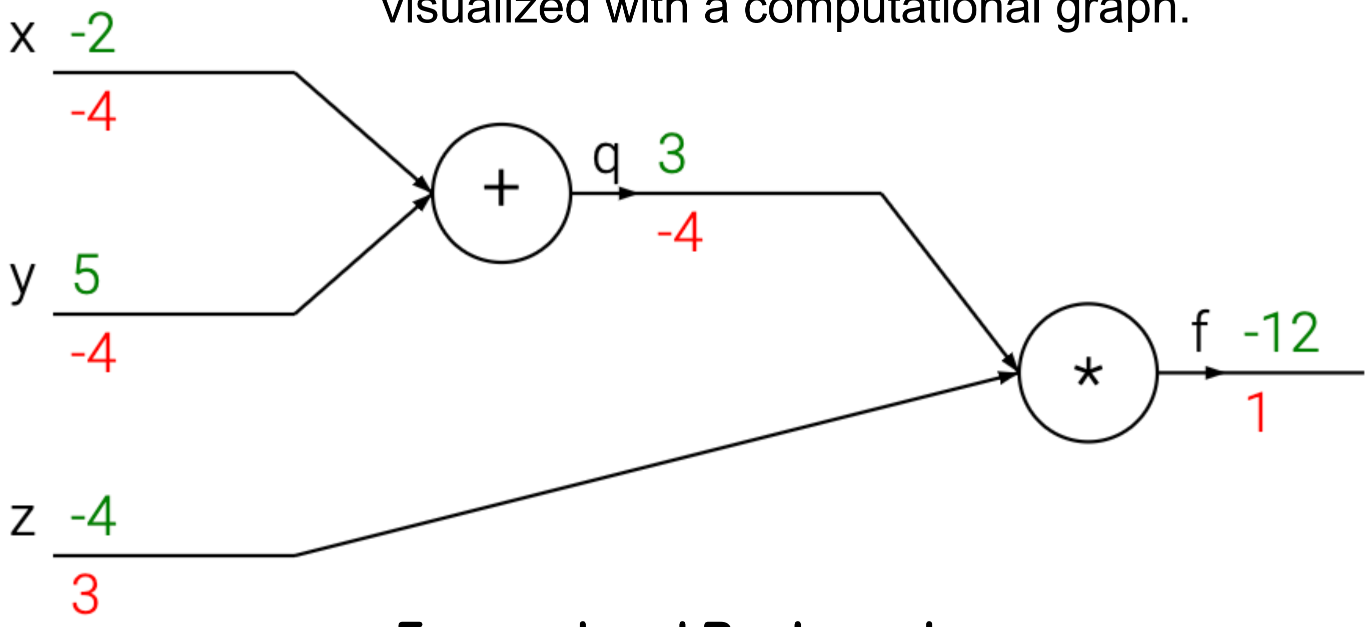
- Knowing how to back-propagate gradients = Knowing how to compute gradients on ANY neural networks!!!

6. Backpropagation

Computational graph

$$q = x + y$$
$$f = q * z$$

This computation can also be nicely visualized with a computational graph.



Forward and Backward passes

6. Backpropagation

AIBYNETO

1D Backpropagation example

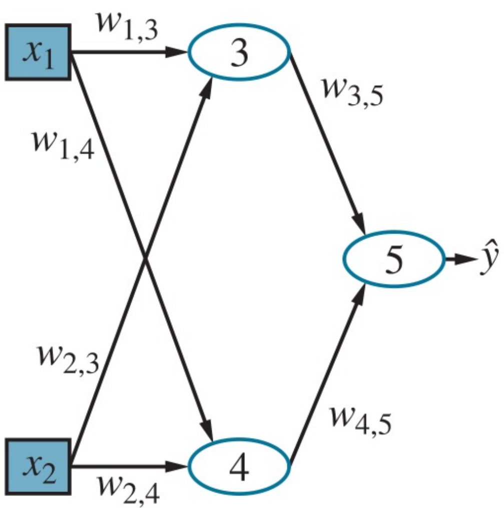
Suppose $a = wx \approx y$.

Given $x = 2, w = 2.5$, and $y = 3$, what is the new squared loss value after one update with a step size of 0.1?

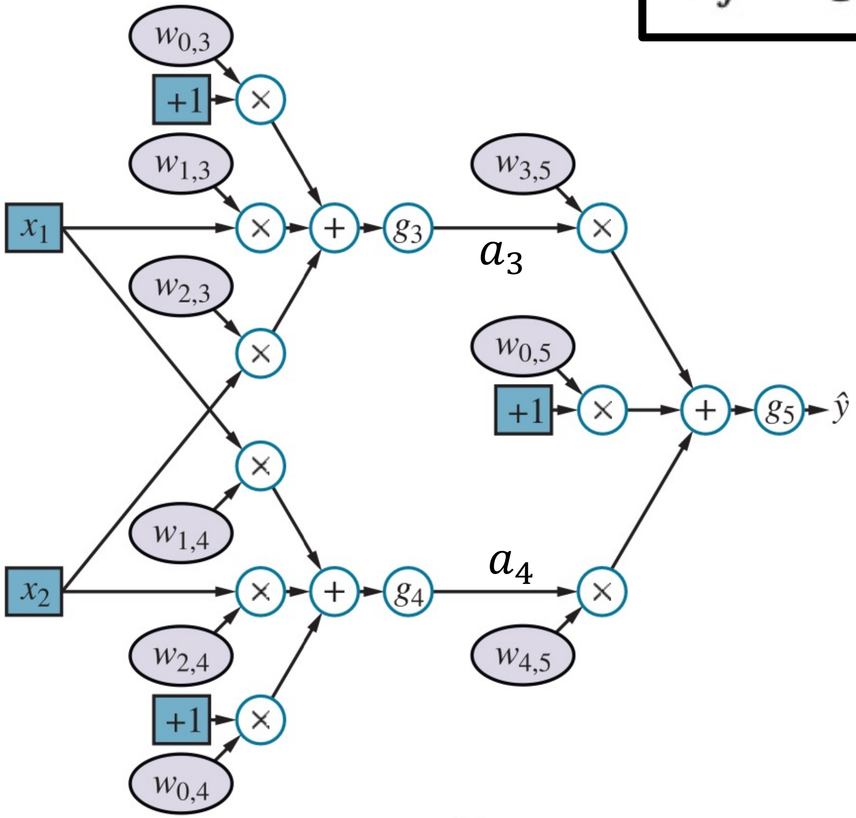
Extra

AIBYNETO

$$a_j = g_j\left(\sum_i w_{i,j} a_i\right) \equiv g_j(in_j)$$

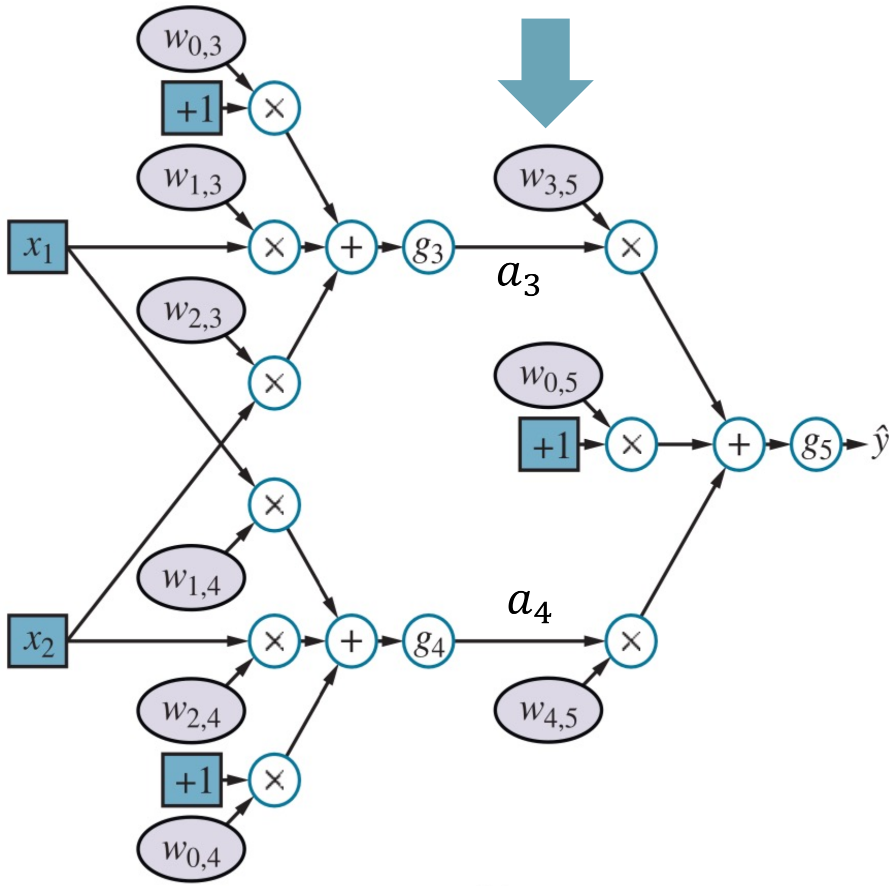


(a)



(b)

Extra

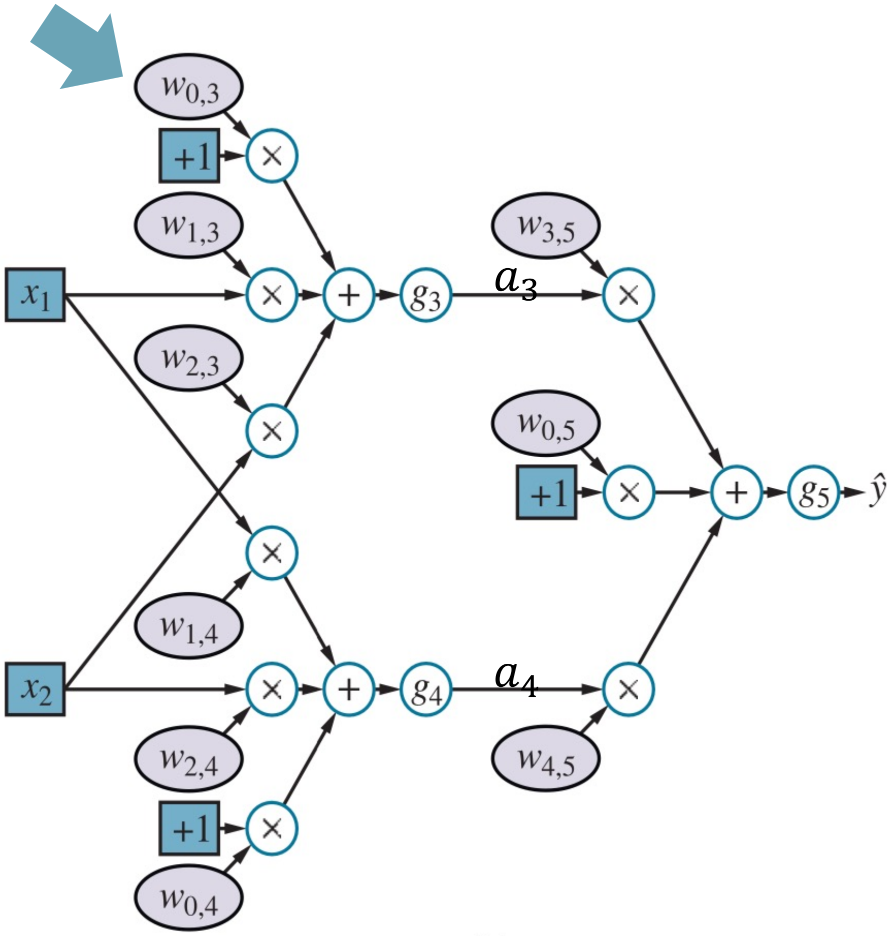


(b)

$$\boxed{a_j = g_j\left(\textstyle\sum_i w_{i,j} a_i\right) \equiv g_j(in_j)}$$

$$\begin{aligned}
\frac{\partial}{\partial w_{3,5}} Loss(h_{\mathbf{w}}) &= \frac{\partial}{\partial w_{3,5}}(y-\hat{y})^2 = -2(y-\hat{y})\frac{\partial \hat{y}}{\partial w_{3,5}} \\
&= -2(y-\hat{y})\frac{\partial}{\partial w_{3,5}} g_5(in_5) = -2(y-\hat{y})g_5'(in_5)\frac{\partial}{\partial w_{3,5}} in_5 \\
&= -2(y-\hat{y})g_5'(in_5)\frac{\partial}{\partial w_{3,5}}\left(w_{0,5} + w_{3,5}a_3 + w_{4,5}a_4\right) \\
&= -2(y-\hat{y})g_5'(in_5)a_3 .
\end{aligned}$$

Extra



(b)

$$\boxed{a_j = g_j\left(\textstyle\sum_i w_{i,j} a_i\right) \equiv g_j(in_j)}$$

$$\begin{aligned}
\frac{\partial}{\partial w_{1,3}} Loss(h_{\mathbf{w}}) &= -2(y-\hat{y})g_5'(in_5)\frac{\partial}{\partial w_{1,3}}\left(w_{0,5} + w_{3,5}\,a_3 + w_{4,5}\,a_4\right) \\
&= -2(y-\hat{y})g_5'(in_5)\,w_{3,5}\,\frac{\partial}{\partial w_{1,3}} a_3 \\
&= -2(y-\hat{y})g_5'(in_5)\,w_{3,5}\,\frac{\partial}{\partial w_{1,3}} g_3(in_3) \\
&= -2(y-\hat{y})g_5'(in_5)\,w_{3,5}\,g_3'(in_3)\,\frac{\partial}{\partial w_{1,3}} in_3 \\
&= -2(y-\hat{y})g_5'(in_5)\,w_{3,5}\,g_3'(in_3)\,\frac{\partial}{\partial w_{1,3}}\left(w_{0,3} + w_{1,3}\,x_1 + w_{2,3}\,x_2\right) \\
&= -2(y-\hat{y})g_5'(in_5)\,w_{3,5}\,g_3'(in_3)\,x_1\,.
\end{aligned}$$

Follow AIBYNETO

AIBYNETO

[AI Powered by Neto-san]

บทความฟรี

[AIBYNETO]

คอร์สฟรี

[AIBYNETO]

ชีทเรียน + code ฟรี